

FIG. 1

C SOURCE CODE
N=8;
for(i=0; i<N; i++)
 y+=x[i];

#1.	LDRS _repeat_start
#2.	LDRE _repeat_end
#3.	LDRC #8
#4.	MOV #x_addr,r0
#5.	MOV #y_addr,r1
#6.	MOV #0,r3
#7. _repeat_start	MOV @r0+,r2
#8. _repeat_end	ADD r2,r3
#9.	MOV r3,@r1

FIG. 2

LDRS _repeat_start : MAKE REPEAT START PC _repeat_start
LDRE _repeat_end : MAKE REPEAT END PC _repeat_end
LDRC #N : MAKE REPEAT NUMBER N

FIG. 3

I	D0	E				
	D1	L0	L1	L2	L3	

FIG. 6

#1.LDRS _repeat_start	#7.MOV @r0+,r2	#11._repeat_start	ADD r2,r3	#19.ADD r2,r3
#2.LDRE _repeat_end	#8.MOV @r0+,r4	#12.	MOV @r0+,r2	#20.ADD r4,r3
#3.LDRC #2	#9.MOV @r0+,r5	#13.	ADD r4,r3	#21.ADD r5,r3
#4.MOV #x_addr,r0	#10.MOV @r0+,r6	#14.	MOV @r0+,r4	#22.ADD r6,r3
#5.MOV #y_addr,r1		#15.	ADD r5,r3	#23.MOV r3,@r1
#6.MOV #0,r3		#16.	MOV @r0+,r5	
		#17.	ADD r6,r3	
		#18._repeat_end	MOV @r0+,r6	

FIG. 7

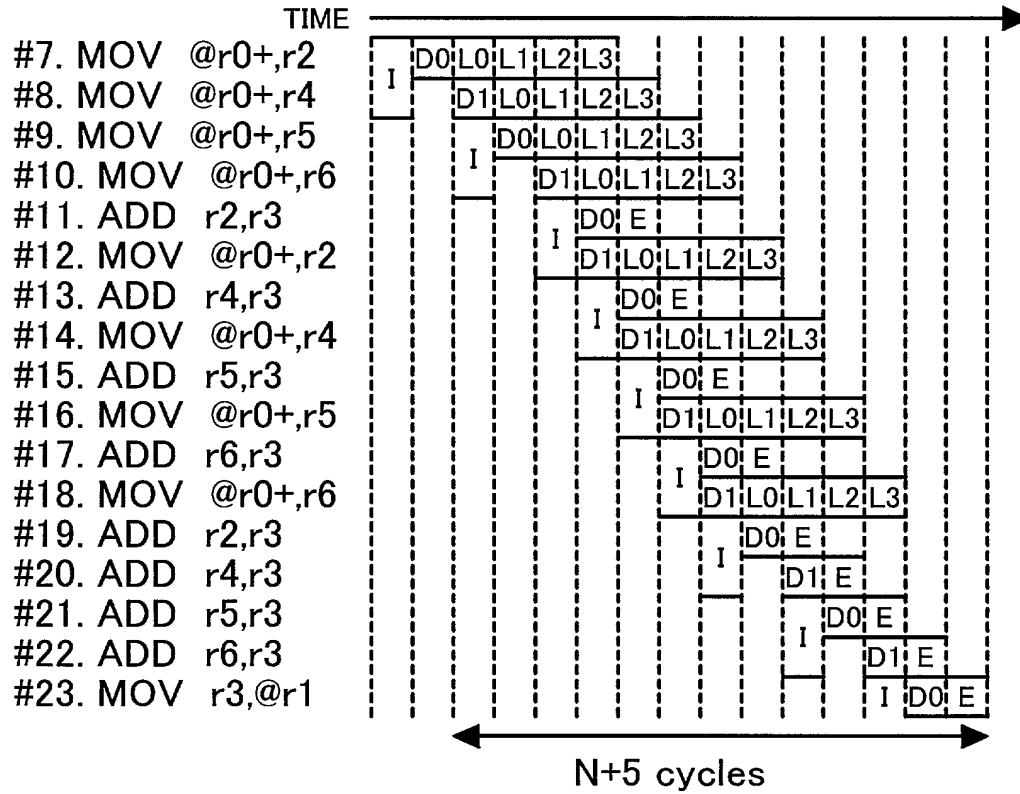


FIG. 8

#1.	MOV #8,r4	#5.MOV @r0+,r2	#9. TERM
#2.	MOV #x_addr,r0	#6.ADD r2,r3	#10._exit THABORT
#3.	MOV #y_addr,r1	#7.DT r4	#11. MOV r3,@r1
#4._thread0	FORK_thread0	#8.BT/S _exit	

FIG. 9

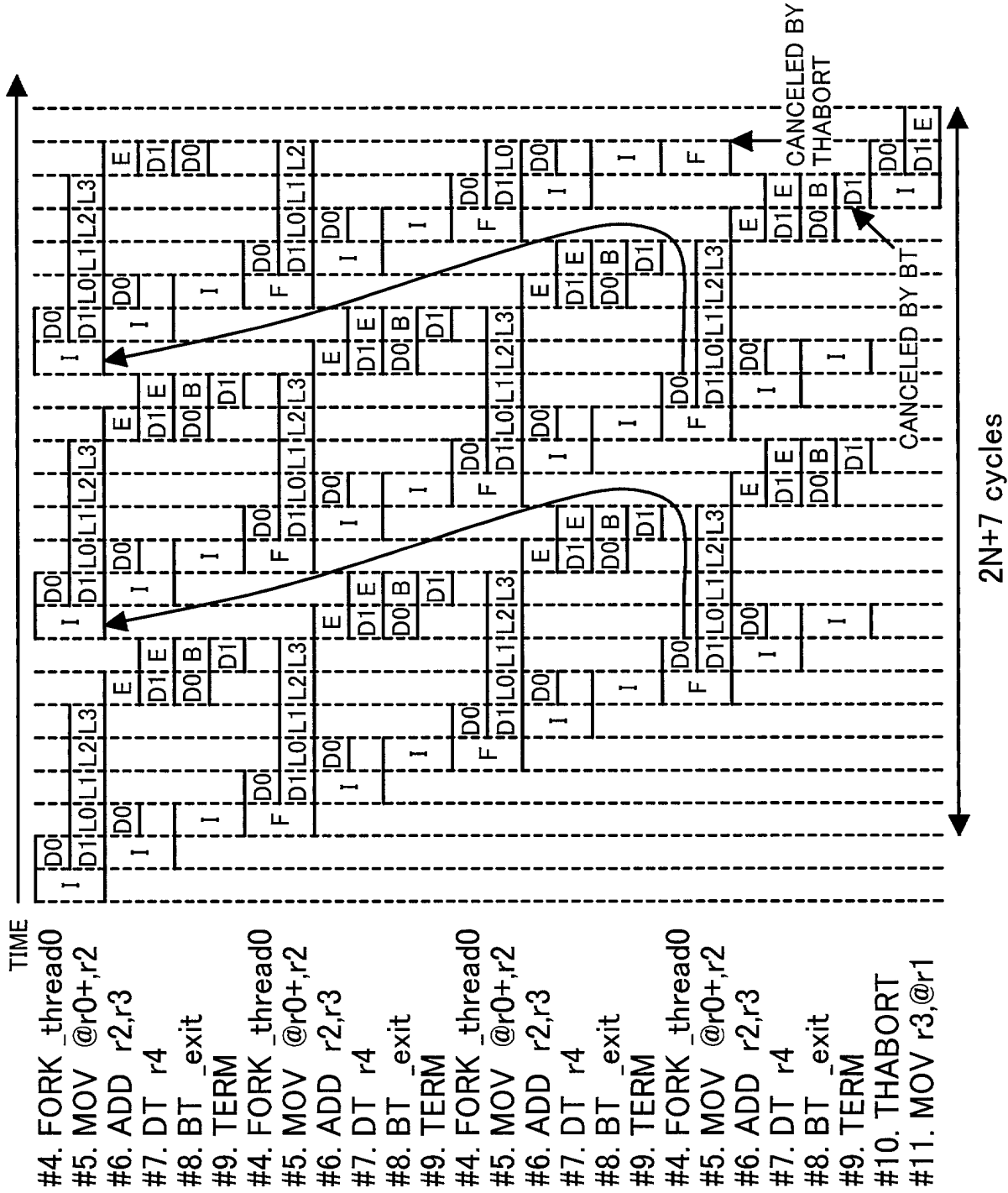


FIG. 10

#1.LDRS _repeat_start	#7. _thread0	MOV @r0+,r2
#2.LDRE _repeat_end	#8.	UNCOND_SUSPEND
#3.MOV #x_addr,r0	#9. _repeat_start	FORK _thread1
#4.FORK thread0	#10.	ADD r2,r3
#5.LDRC #7	#11. _thread1	MOV @r0+,r2
#6.MOV #y_addr,r1	#12. _repeaat_end	UNCOND_SUSPEND
	#13.	ADD r2,r3
	#14.	MOV r3,@r1

FIG. 11

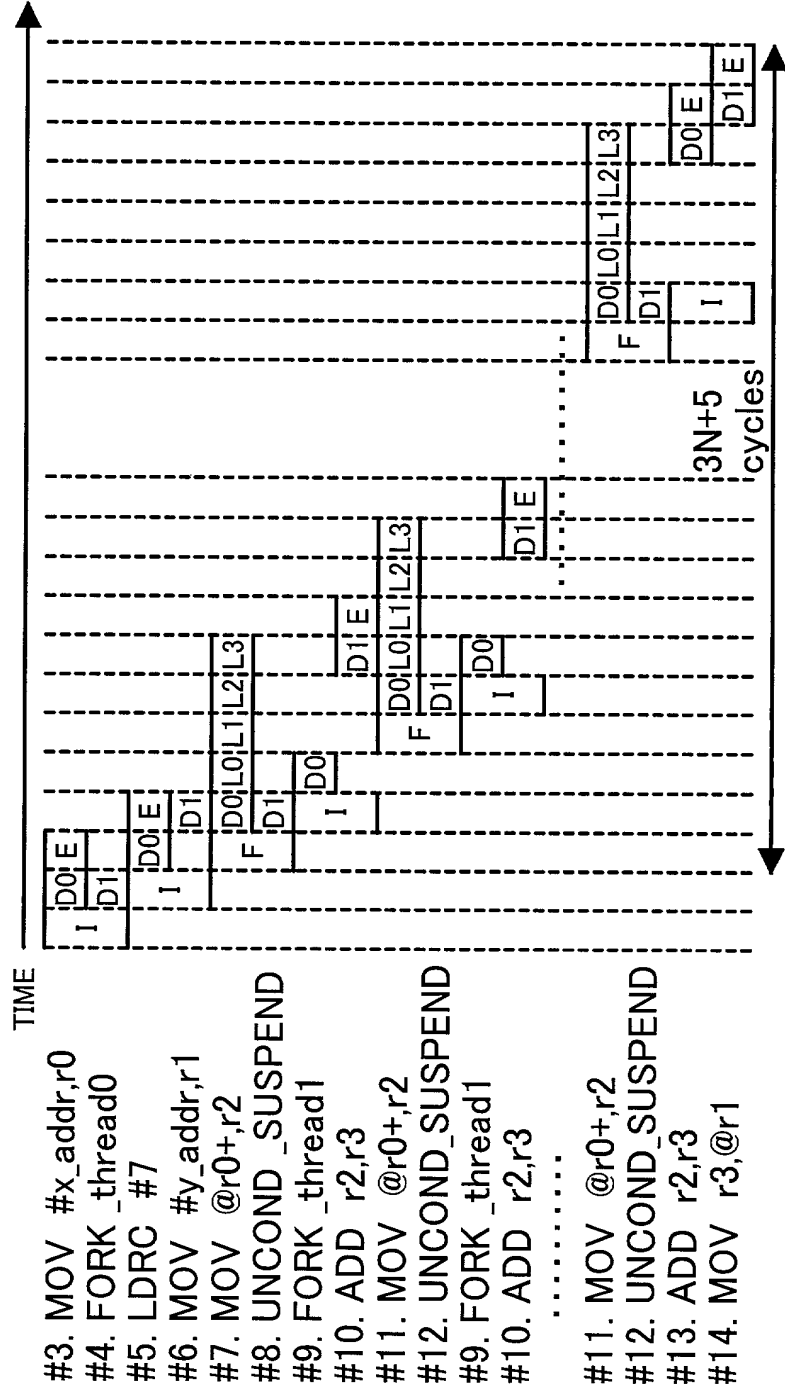


FIG. 12

	DATA NUMBER	LOAD LATENCY	OUT OF ORDER, SOFTWARE PIPELINE	MERLOT METHOD	PRIOR ART (JPA8-249183)	CONVENTIONAL PROCESSOR
#1	N	L	$N+L+1$	$\text{MAX}(2N+L+2, (L+3)N/4+7)$	$\text{MAX}(3N+L+1, (L-1)N+5)$	$LN+2$
#2		4	$N+5$	$2N+7$	$3N+5$	$4N+2$
#3		30	$N+31$	$33N/4+7$	$29N+5$	$30N+2$
#4	8	4	13	23	29	34
#5		30	39	73	237	242
#6	32	4	37	51	101	130
#7		30	63	271	933	962

FIG. 13

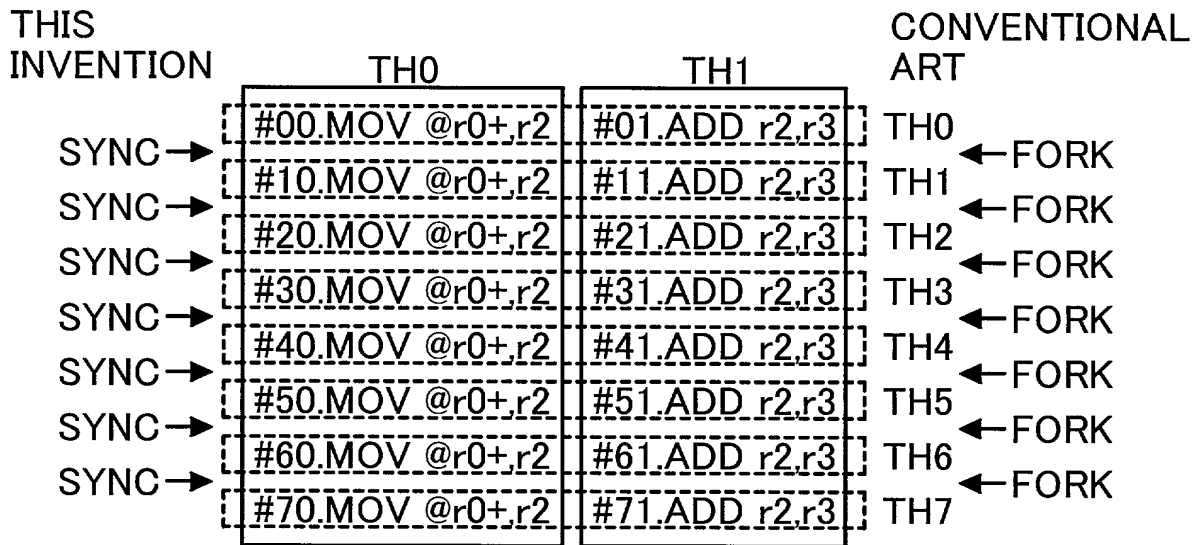


FIG. 14

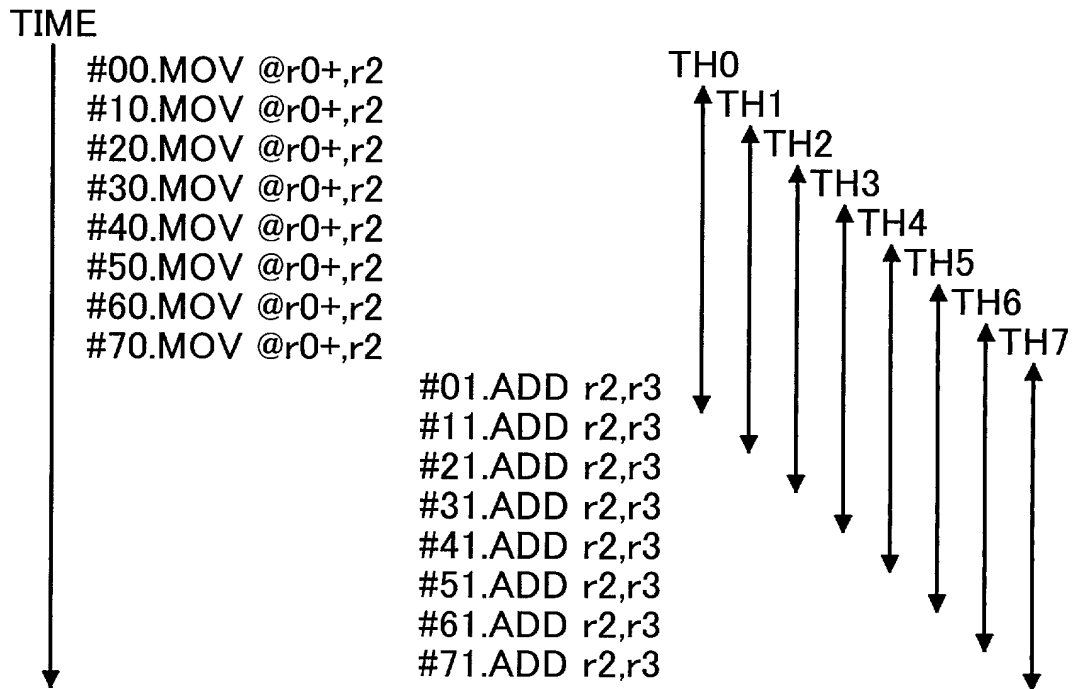


FIG. 15

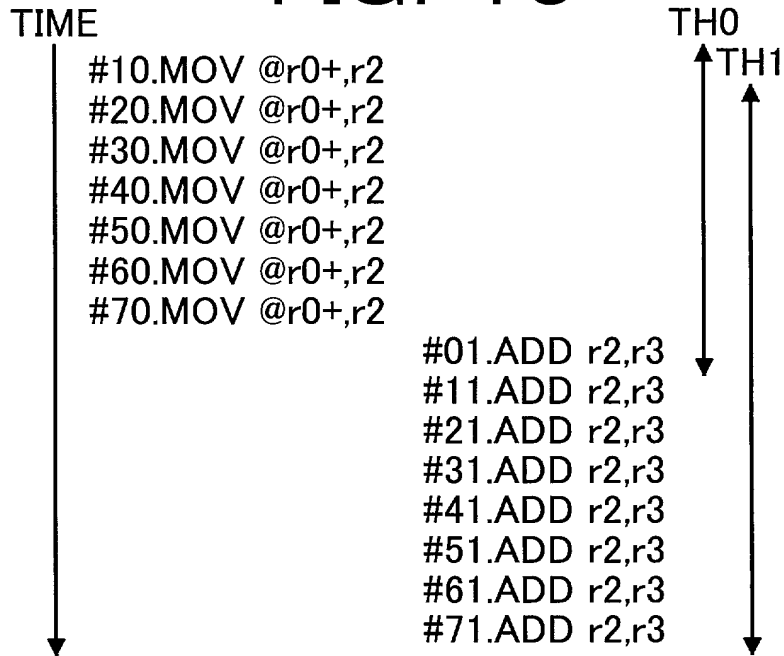


FIG. 16

#1.	LDRE _repeat0	#11._thred1	LDRS _repeat1
#2.	MOV #x_addr,r0	#12.	LDRE _repeat1
#3.	LDRS _repeat0	#13.	LDRC #8
#4.	MOV #y_addr,r1	#14.	NOP
#5.	THRDG/R _thred1	#15._repeat1	ADD r2,r3
#6.	MOV #0,r3	#16.	MOV r3,@r1
#7.	LDRC #8	#17.	THRDE
#8.	NOP		
#9._repeat0	MOV @r0+,r2		
#10.	SYNCE		

THRDG/R: THREAD GENERATION REPEAT TYPE
THRDE : THREAD END
SYNCE : THREAD END SYNCHRONISM

FIG. 17



Year	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100
1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	

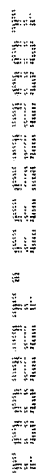


FIG. 19

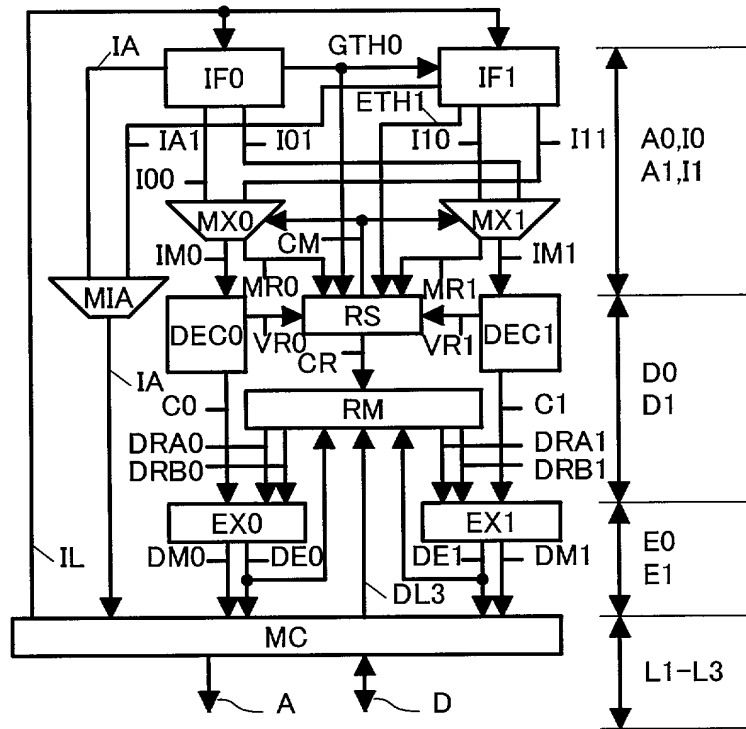


FIG. 20

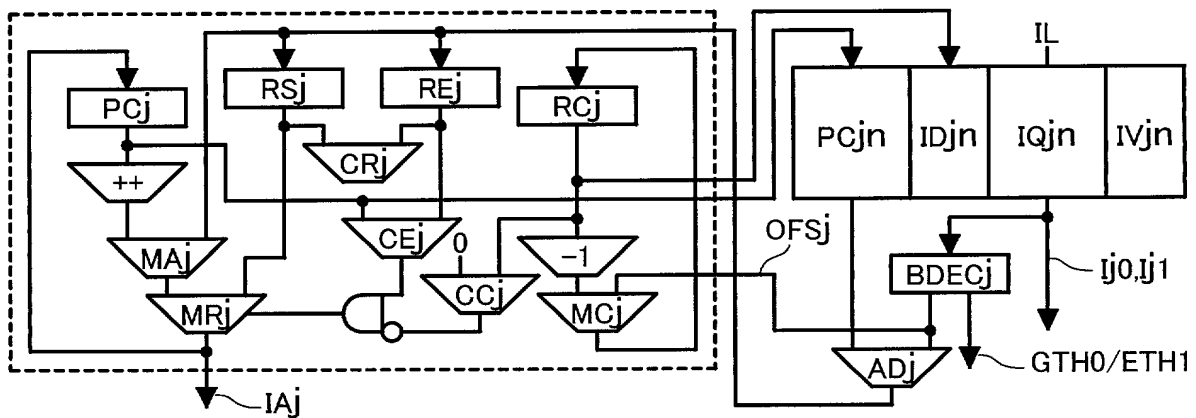


FIG. 21

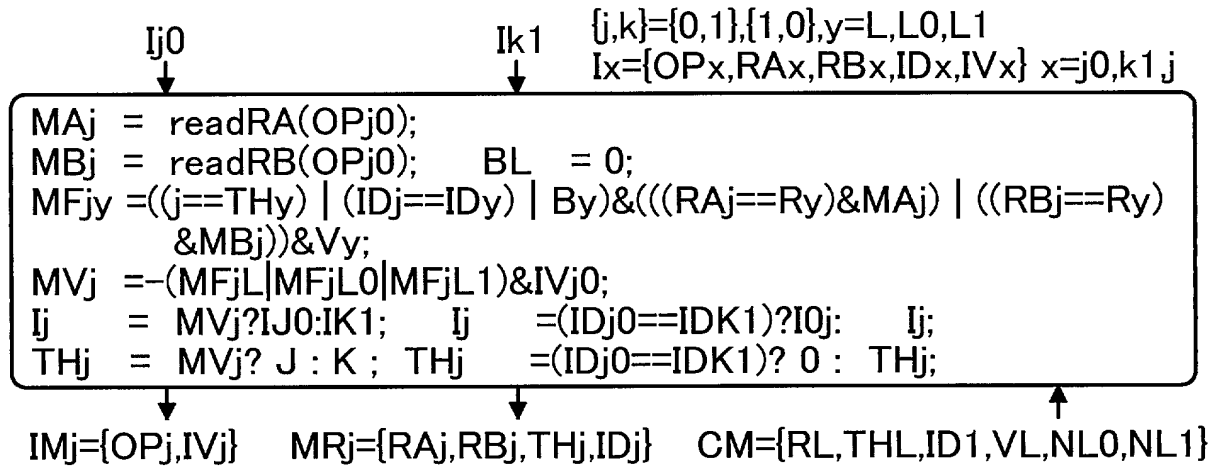


FIG. 22

	EXECUTING ENABLE OR DISABLE		SELECTING INSTRUCTION	
	I00	I10	I0	I1
#1	ENABLE	ENABLE	I00	I10
#2	ENABLE	DISABLE	I00	I01
#3	DISABLE	ENABLE	I11	I10
#4	DISABLE	DISABLE	I11	I01

FIG. 23

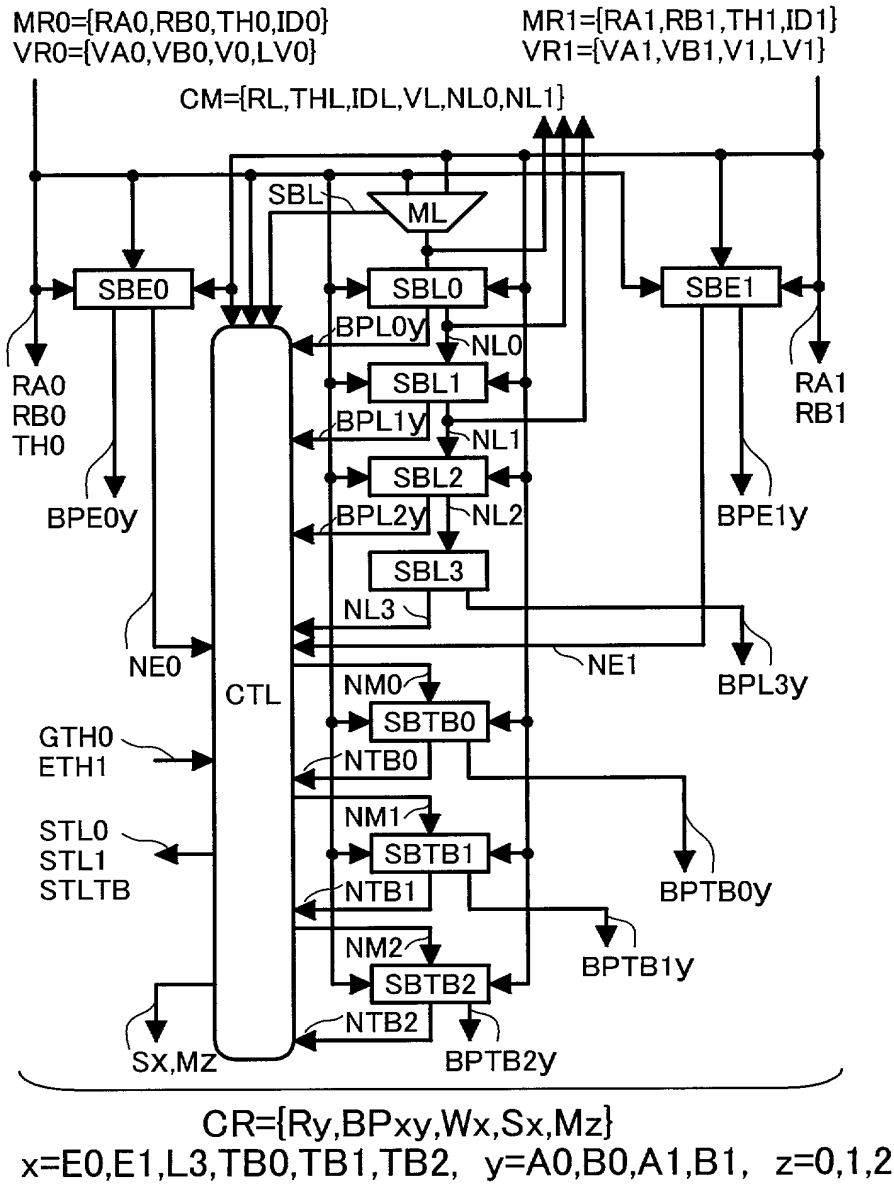


FIG. 24

```

SBL=((TH0==0)&LV0)|((TH0==1)&=-LV1);
if(SBL){RL=RB0;THL=TH0;IDL=ID0;VL=LV0&-STL0}
else  {RL=RB1;THL=TH1;IDL=ID1;VL=LV1&-STL1}

```

FIG. 25

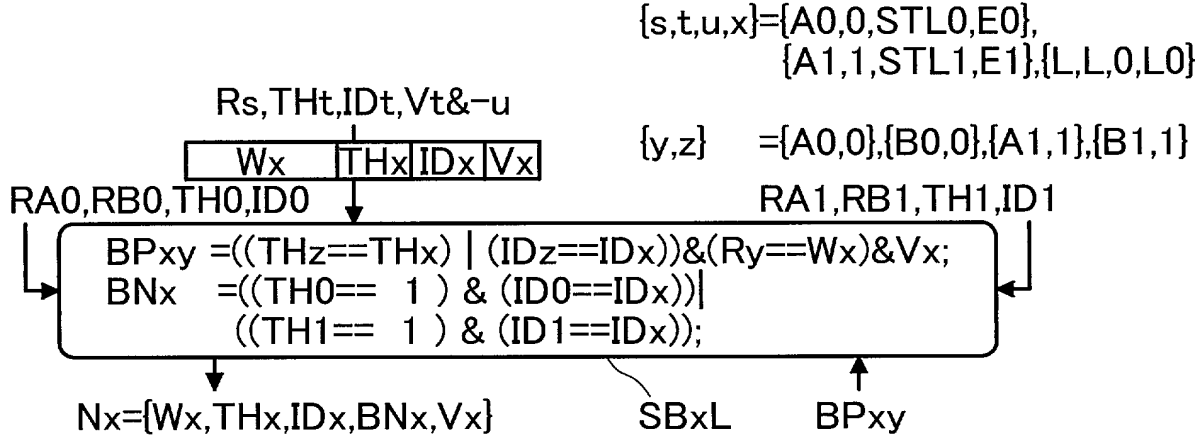


FIG. 26

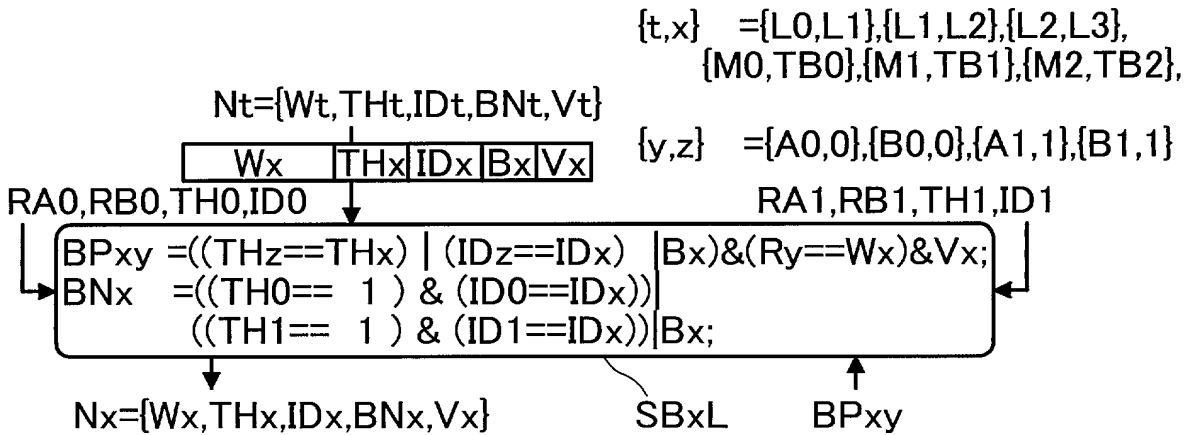


FIG. 27

$STL0 = ((BPL0A0 \mid BPL1A0 \mid BPL2A0) \& VA0) \mid ((BPL0B0 \mid BPL1B0 \mid BPL2B0) \& VB0);$
 $STL1 = ((BPL0A1 \mid BPL1A1 \mid BPL2A1) \& VA1) \mid ((BPL0B1 \mid BPL1B1 \mid BPL2B1) \& VB1);$
 $STL0 \mid= STL1 \& (TH0==1) \mid ((SBL==1) \& LV0);$
 $STL1 \mid= STL0 \& (TH0==0) \mid ((SBL==0) \& LV1);$
 $STH = (-GTH0 \& STH) \mid ETH1;$
 $SX = VX \& ((THx==1) \mid BX \mid STH) \quad \{x=TB0, TB1, TB2, L3, E0, E1\}$
 $CX = VX \& ((THx==0) \& \neg BX \& \neg STH)$

STATE						OUTPUTS			
CTB2	CTB1	CTB0	CL3	CE0	CE1	M2	M1	M0	STLTB
*	*	*	0	0	0	TB2	TB1	TB0	0
*	0	*	0	0	1	TB1	TB0	E1	0
0	*	*							
*	0	*	0	1	0	TB1	TB0	E0	0
0	*	*							
*	0	*	1	0	0	TB1	TB0	L3	0
0	*	*							
0	0	*	0	1	1	TB0	E0	E1	0
			1	0	1	TB0	L3	E1	0
			1	1	0	TB0	L3	E0	0
0	0	0	1	1	1	L3	E0	E1	0
OTHERS						TB2	TB1	TB0	1

$NM2 = (M2==TB2)?NTB2:((M2==TB1)?NTB1:((M2==TB0)?NTB0:((M2==L3)?NL3)));$
 $NM1 = (M1==TB1)?NTB1:((M1==TB0)?NTB0:((M1==L3)?NL3:((M1==E0)?NE0)));$
 $NM0 = (M0==TB0)?NTB0:((M0==L3)?NL3:((M0==E0)?NE0:((M0==E1)?NE1)));$

FIG. 28

$CR=\{R_y, BP_{xy}, W_x, S_x, M_z, TH_0\}$

$(x=E_0, E_1, L_3, TB_0, TB_1, TB_2, y=A_0, B_0, A_1, B_1, z=0, 1, 2)$

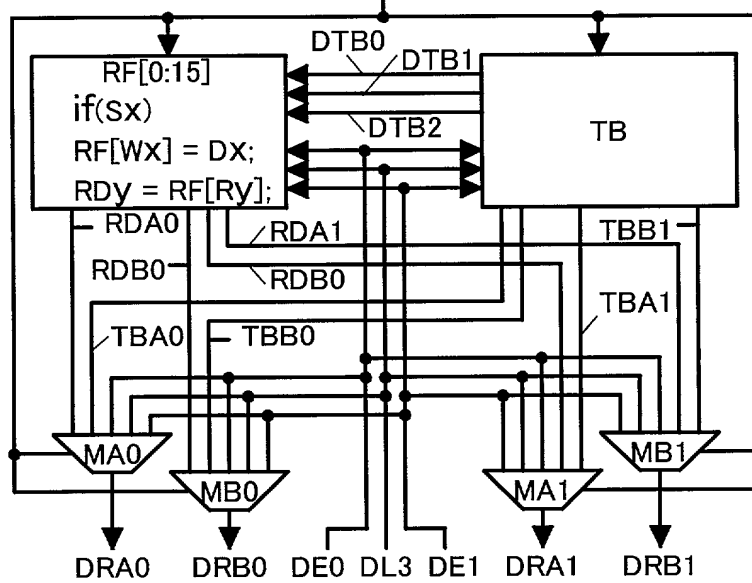


FIG. 29

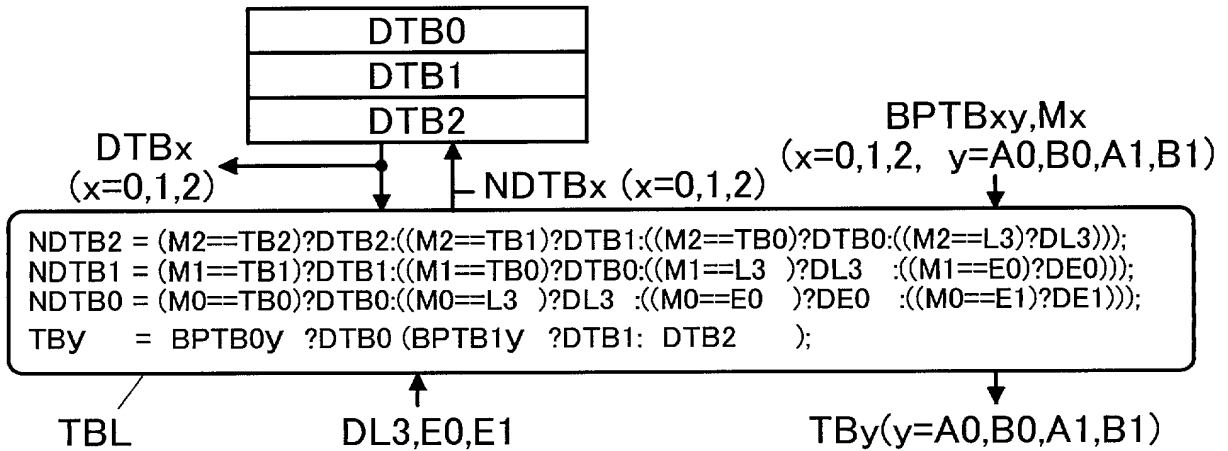


FIG. 30

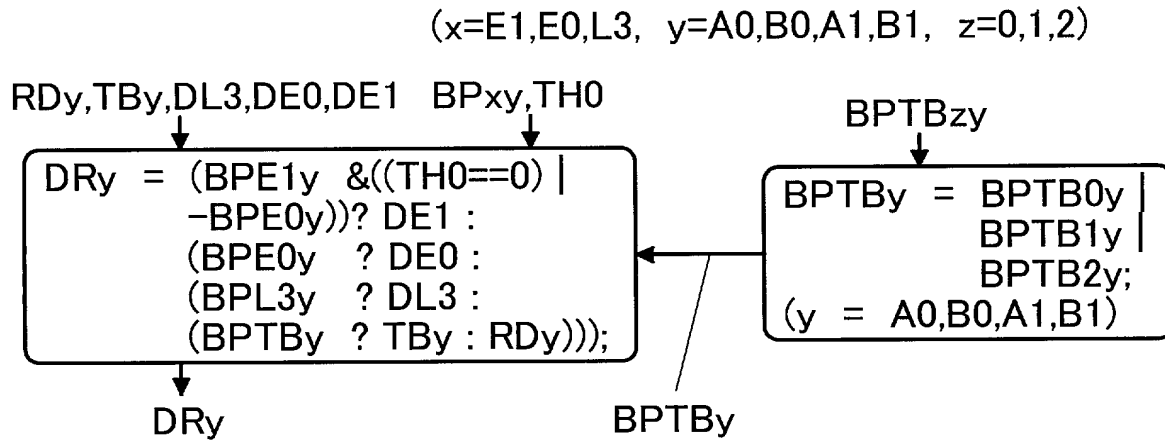


FIG. 31

